



Introduzione

- 0.1 Studio degli algoritmi
- 0.2 Origini delle macchine calcolatrici
- 0.3 Scienza degli algoritmi
- 0.4 Ruolo dell'astrazione
- 0.5 Ripercussioni sociali

L'informatica è la disciplina che cerca di dare un fondamento scientifico ad argomenti come la progettazione di computer, la programmazione, l'elaborazione delle informazioni, le soluzioni algoritmiche dei problemi e il processo algoritmico stesso. Si tratta della disciplina che fornisce le basi scientifiche per tutte le applicazioni, sia attuali sia future. Data la vastità di temi, è impossibile diventare esperti d'informatica studiando soltanto alcuni argomenti isolati, oppure imparando semplicemente a usare gli strumenti di calcolo disponibili oggi. Per capire l'informatica è invece indispensabile cogliere la portata e le dinamiche di un'ampia gamma di argomenti.

Questo libro è progettato proprio per fornire un background di questo tipo: presenta l'informatica attraverso un'introduzione integrata agli argomenti che costituiscono un tipico curriculum informatico universitario; può essere impiegato, quindi, come base per gli studenti che iniziano un corso universitario di studi in informatica oppure costituire un valido supporto per i lettori interessati a un'introduzione alle tecnologie che stanno alla

base della società attuale.

0.1 Studio degli algoritmi

Il libro inizia con il concetto più importante dell'informatica, quello di algoritmo. Informalmente, un **algoritmo** è la sequenza di passi che definiscono il modo in cui viene eseguita un'operazione¹. Per esempio esistono algoritmi per costruire aeromodelli (sotto forma di fogli di istruzioni), per far funzionare lavatrici (di solito si trovano all'interno degli obblò), per eseguire brani musicali (sotto forma di spartiti) e per fare trucchi di magia (Figura 0.1).

Prima che un dispositivo possa eseguire una qualsiasi operazione, bisogna individuare un algoritmo per effettuarla e rappresentare tale algoritmo in una forma compatibile con la macchina. La rappresentazione di un algoritmo compatibile con la macchina è definita **programma**. I programmi, e gli algoritmi che rappresentano, sono chiamati globalmente **software**, in contrapposizione ai dispositivi fisici, detti **hardware**.

Effetto: il 'mago' sceglie alcune carte da un normale mazzo di carte da gioco, le dispone a faccia in giù su un tavolo e mischiandole le colloca in ordine sparso. Gli spettatori dicono quale carta desiderano, se rossa o nera, e il mago ne gira una del colore richiesto.

Segreto e terminologia

- Passo 1** Da un normale mazzo di carte si scelgano dieci carte rosse e poi dieci carte nere. Le si disponga sul tavolo a faccia in su, divise in due mazzi secondo il colore.
- Passo 2** Si dichiari di aver scelto alcune carte rosse e alcune nere.
- Passo 3** Si raccolgano le carte rosse. Fingendo di allinearle in un piccolo mazzo, le si tenga a faccia in giù nella mano sinistra, e con il pollice e l'indice della mano destra si tirino le estremità del mazzo in modo che ogni carta sia leggermente curvata all'indietro. Si ponga il mazzo di carte rosse a faccia in giù sul tavolo mentre si dice: "Ecco le carte rosse in questo mazzo".
- Passo 4** Si raccolgano le carte nere. Procedendo come nel Passo 3, le si pieghi in avanti, poi le si rimetta sul tavolo a faccia in giù dicendo: "Ed ecco in questo mazzo le carte nere".
- Passo 5** Subito dopo aver rimesso le carte nere sul tavolo, si usino ambedue le mani per mescolare le carte rosse e nere (ancora a faccia in giù) mentre le si distribuisce sopra il tavolo. Si spieghi che si stanno mescolando le carte.
- Passo 6** Finché ci sono ancora carte a faccia in giù sul tavolo, si eseguano ripetutamente i passi seguenti:
- 6.1 si domandi al pubblico di richiedere una carta rossa o una nera
 - 6.2 se il colore richiesto è rosso e c'è una carta a faccia in giù piegata verso l'interno, la si giri dicendo: "Ecco una carta rossa"
 - 6.3 se il colore richiesto è nero e c'è una carta a faccia in giù piegata verso l'esterno, la si giri dicendo: "Ecco una carta nera"
 - 6.4 altrimenti, si affermi che non ci sono più carte del colore richiesto e si girino quelle che restano per dimostrare che si ha ragione.

Figura 0.1 Un algoritmo per un gioco di prestigio.

¹ Più precisamente, un algoritmo è una sequenza ordinata di passi non ambigui ed eseguibili, che definiscono un'attività di lunghezza finita. Questi dettagli saranno trattati nel Capitolo 4.

Descrizione: questo algoritmo riceve in input due interi positivi e ne calcola il massimo comune divisore.

Procedura

- Passo 1** Si assegni a M e a N, rispettivamente, il valore maggiore e minore dei due valori di ingresso.
- Passo 2** Si divida M per N, e si denomi il resto R.
- Passo 3** Se R non è 0, si assegni a M il valore di N, si assegni a N il valore di R, e si ritorni al Passo 2; altrimenti, il massimo comune divisore è il valore assegnato a N.

Figura 0.2 Algoritmo di Euclide per calcolare il massimo comune divisore di due interi positivi.

Lo studio degli algoritmi ha avuto inizio in ambito matematico. In realtà, la ricerca di algoritmi efficienti era un'attività importante dei matematici già molto tempo prima dello sviluppo dei computer odierni. L'obiettivo principale era trovare un'unica sequenza di istruzioni che descrivessero come risolvere tutti i problemi di un particolare tipo. Uno degli esempi più noti è l'algoritmo della divisione, usato per calcolare il quoziente di due numeri di più cifre. Un altro esempio è l'algoritmo euclideo, scoperto dal matematico greco Euclide, per calcolare il massimo comune divisore di due interi positivi (Figura 0.2).

Una volta trovato un algoritmo per eseguire un determinato compito, svolgerlo non richiede più la conoscenza dei principi su cui si basa l'algoritmo: basta seguire pedissequamente le istruzioni che lo compongono. (Si può usare l'algoritmo della divisione per calcolare un quoziente oppure l'algoritmo di Euclide per trovare il massimo comune divisore tra due numeri senza capirne il funzionamento). In un certo senso, l'intelligenza necessaria per risolvere il problema è codificata nell'algoritmo.

È attraverso questa capacità di catturare e convogliare intelligenza tramite gli algoritmi che vengono create macchine dal comportamento intelligente; di conseguenza, il livello d'intelligenza delle macchine è limitato a quanta può esserne fornita loro mediante gli algoritmi. Solo se troviamo un algoritmo che guida l'esecuzione di un'operazione possiamo creare una macchina per automatizzarla. D'altra parte, se non esiste alcun algoritmo per risolvere un problema, questo significa che la sua soluzione va al di là delle capacità delle macchine.

Lo sviluppo di algoritmi è quindi un obiettivo fondamentale dell'informatica, che proprio a tale argomento dedica una parte significativa. Oltre a ciò, è possibile rendersi conto della vastità di questa disciplina considerando altri aspetti. Uno di essi riguarda il modo in cui vengono scoperti gli algoritmi, una questione strettamente collegata al discorso del *problem solving*. Individuare un algoritmo per risolvere un problema equivale a risolvere il problema stesso. Ne segue che gli studi in questo settore dell'informatica sono molto vincolati agli sviluppi della ricerca in altre aree, come la psicologia del *problem solving* negli esseri umani e le teorie dell'istruzione e dell'apprendimento. Alcuni di questi concetti verranno trattati nel Capitolo 4.

Una volta scoperto un algoritmo per risolvere un problema, la fase successiva è riprodurlo in una forma che possa essere comunicata a una macchina o ad altre persone. Ciò significa che lo si deve trasformare a livello concettuale in una chiara sequenza di istruzioni, e significa anche che occorre rappresentare queste istruzioni in modo non ambiguo. Gli studi in quest'ambito attingono dalle nostre conoscenze sul linguaggio e sulle grammatiche, e hanno portato alla messa a punto di **numerosi schemi di rappresentazione degli algoritmi, i linguaggi di programmazione**, che si basano su vari approcci al processo di programmazione, definiti *paradigmi di programmazione*. Alcuni di questi linguaggi e i relativi paradigmi di programmazione saranno esaminati nel Capitolo 5.

Man mano che la tecnologia informatica è stata applicata a problemi sempre più complicati, i ricercatori hanno capito che il progetto di grandi sistemi software è molto di più dello sviluppo di singoli algoritmi. Richiede, infatti, anche la progettazione dell'interazione tra i vari componenti. Per gestire questa complessità, gli informatici hanno fatto riferimento al campo ben definito dell'ingegneria, nella speranza di trovare gli strumenti adatti. Il risultato è il settore dell'informatica noto come ingegneria del software, che oggi prende spunto da diversi campi: l'ingegneria, la gestione dei progetti e del personale, e la progettazione dei linguaggi di programmazione. L'ingegneria del software verrà trattata nel Capitolo 6.

Un'altra area importante dell'informatica concerne il progetto e la costruzione di macchine per eseguire gli algoritmi. Questi argomenti saranno trattati nei Capitoli 1 e 2. Sebbene lo studio della struttura dei computer richieda alcune discussioni sulla tecnologia moderna, questo libro non si propone di approfondire i dettagli sul modo in cui le odierne architetture di calcolo sono realizzate a livello circuitale; questo, infatti, ci spingerebbe troppo dentro il campo dell'ingegneria elettronica. Inoltre, proprio come i calcolatori meccanici del passato hanno ceduto il passo ai dispositivi elettronici, l'elettronica contemporanea potrebbe essere sostituita in futuro da altre tecnologie, l'ottica innanzitutto. Il nostro obiettivo qui è semplicemente quello di capire quanto basta della tecnologia da poterne apprezzare le derivazioni nelle macchine attuali e anche l'influsso sullo sviluppo dell'informatica del futuro.

In teoria, l'architettura dei computer dovrebbe essere una conseguenza diretta delle nostre conoscenze dei processi algoritmici a prescindere dalle disponibilità tecnologiche. In altre parole, anziché consentire ai vincoli tecnologici di guidare il progetto delle macchine, e quindi il modo di rappresentare gli algoritmi, dovrebbero essere le nostre conoscenze sugli algoritmi a fare da forza trainante nella definizione delle architetture dei calcolatori. Via via che la tecnologia progredisce, questo sogno diventa sempre più realtà. Oggi è possibile costruire macchine che permettono di rappresentare gli algoritmi come sequenze di istruzioni multiple eseguite simultaneamente oppure come schemi di collegamento tra molte unità di elaborazione, nello stesso modo in cui la mente umana rappresenta le informazioni come collegamenti tra i neuroni (Capitolo 10).

L'architettura dei computer viene studiata anche nel contesto relativo alla memorizzazione e al recupero dei dati. In questi casi le funzioni interne di un computer si riflettono spesso nelle sue caratteristiche esterne; queste peculiarità e i modi per evitarne gli effetti indesiderati vengono trattati nel Capitolo 1, 7, 8 e 9.

Strettamente connesso al progetto delle macchine calcolatrici è il progetto della loro interfaccia con il mondo esterno. Per esempio: come inserire gli algoritmi in una macchina e come dirle quale algoritmo eseguire? La soluzione di questo problema in un ambiente in cui una singola macchina deve fornire servizi diversi richiede di affrontare preventivamente parecchi altri problemi relativi al coordinamento delle attività e alla allocazione delle risorse. Alcune soluzioni vengono esaminate nel Capitolo 3, nell'ambito della descrizione dei sistemi operativi.

Poiché si richiede alle macchine di eseguire compiti sempre più intelligenti, l'informatica si è indirizzata anche allo studio dell'intelligenza umana. La speranza è che capendo come ragionano e comprendono le menti degli uomini si possa progettare algoritmi che riproducono questi processi, trasferendo quindi tali capacità alle macchine. Il risultato di queste ricerche è la parte dell'informatica nota come intelligenza artificiale, rivolta soprattutto ad aree di ricerca come la psicologia, la biologia e la linguistica. Alcuni di questi argomenti sono trattati nel Capitolo 10.

La ricerca di algoritmi per risolvere problemi sempre più complessi pone alcune questioni sui **limiti dei processi algoritmici**. Se non esiste alcun algoritmo idoneo a un dato problema, quel problema non può essere risolto da una macchina; cioè i computer sono in grado di trovare una soluzione solo a problemi risolvibili algoritmicamente.

Il fatto che esistano problemi privi di una soluzione algoritmica è emerso come tema di ricerca in ambito matematico negli anni Trenta, con la pubblicazione del teorema di incompletezza di Kurt Gödel. Questo teorema asserisce nella sostanza che in qualsiasi teoria matematica, che includa il sistema dell'aritmetica, esistono affermazioni che non possono essere provate né confutate. In breve, un qualsiasi studio completo dell'aritmetica va oltre le possibilità degli algoritmi.

Il desiderio di indagare i limiti dei metodi algoritmici, seguito alla scoperta di Gödel, ha portato i matematici a progettare macchine astratte per eseguire algoritmi (ben prima che la tecnologia fosse in grado di fornire strumenti di calcolo concreti per questo tipo di ricerche) e studiarne le potenzialità teoriche. Attualmente lo studio degli algoritmi e delle macchine astratte costituisce la spina dorsale teorica dell'informatica. Alcuni di questi argomenti sono trattati nel Capitolo 11.

0.2 Origini delle macchine calcolatrici

Le macchine astratte immaginate dai matematici nei primi anni del Novecento rappresentano una parte importante dell'albero genealogico dei computer di oggi, ma altre derivazioni risalgono a molto tempo prima. In realtà, la ricerca sulle macchine che eseguono compiti algoritmici ha una lunga storia.

Uno dei primi dispositivi di calcolo fu l'abaco: la sua origine viene fatta risalire alle civiltà dei Greci e dei Romani. Si tratta di un arnese piuttosto semplice, che consiste di perline forate infilate su bastoncini, che a loro volta sono montati su un telaio rettangolare. Quando le perline vengono spostate lungo i bastoncini, le loro posizioni rappresentano i valori memorizzati. In altre parole è nelle posizioni delle perline che questo 'com-

poter rappresentare e memorizzare i dati. L'abaco è solo uno strumento di memorizzare di dati; per il controllo dell'esecuzione di un algoritmo, lo strumento si basa sull'operatore umano: per creare una macchina calcolatrice completa dev'essere abbinato a una persona.

In anni più recenti la progettazione delle macchine calcolatrici si è basata sulla tecnologia degli **ingranaggi**. Scienziati e inventori attivi in questo campo furono il francese Blaise Pascal (1623-1662), il tedesco Gottfried Wilhelm **Leibniz** (1646-1716) e l'inglese Charles Babbage (1792-1871). I loro dispositivi rappresentavano i dati mediante la posizione di ingranaggi, e i dati erano inseriti meccanicamente impostando le posizioni iniziali degli ingranaggi stessi. Per raccogliere i risultati dei calcoli dei dispositivi di Pascal e di Leibniz bisognava osservare le posizioni finali degli ingranaggi, nello stesso modo in cui si leggono i numeri sul contachilometri di un'automobile. Babbage, dal canto suo, prese in considerazione una macchina per la stampa su carta dei risultati (che chiamò macchina analitica), eliminando così la possibilità di errori di trascrizione.

Per quanto riguarda le capacità di esecuzione di un algoritmo, **nei vari dispositivi descritti finora si può osservare un aumento di flessibilità.** La macchina di Pascal fu costruita per eseguire solo l'algoritmo di addizione, e pertanto la sequenza di passi appropriata fu incorporata nella struttura stessa della macchina. Allo stesso modo, la macchina di Leibniz aveva i suoi algoritmi saldamente inglobati nell'architettura, sebbene offrisse diverse operazioni aritmetiche tra cui scegliere. La macchina alle differenze di Babbage (di cui fu costruito solo un modello dimostrativo) poteva invece essere modificata per eseguire vari tipi di calcolo. La macchina analitica (per la cui costruzione Babbage non ricevette mai alcun finanziamento) fu progettata per leggere le istruzioni sotto forma di fori su schede di cartone; ragion per cui risultava programmabile. La sua sostenitrice, **Augusta Ada Byron**, è spesso riconosciuta oggi come la prima programmatrice del mondo.

L'idea di codificare un algoritmo mediante dei fori su cartone non fu, però, di Babbage. Egli la mutuò da Joseph Jacquard (1752-1834), che nel 1801 aveva sviluppato un telaio in cui le fasi del processo di tessitura erano determinate da configurazioni di fori su schede di cartone (Figura 0.3). In questo modo l'algoritmo seguito dal telaio poteva essere modificato senza difficoltà per ottenere tessuti a disegni diversi. Un altro beneficiario dell'idea di Jacquard fu Herman Hollerith (1860-1929), che applicò il concetto di rappresentazione delle informazioni come fori su schede di cartone per velocizzare il processo di elaborazione durante il censimento degli Stati Uniti nel 1890. (Fu questo lavoro di Hollerith che portò alla nascita di IBM). Alla fine queste schede di cartone arrivarono a essere universalmente conosciute come **schede perforate** e sopravvissero in qualità di strumento di comunicazione con i computer fino agli anni Settanta. Tale tecnica è in realtà attuale ancora oggi, come hanno dimostrato i problemi relativi al conteggio dei voti emersi durante le elezioni presidenziali americane del 2000.

La tecnologia di quei tempi non fu in grado di produrre in maniera economicamente accettabile le complesse macchine a ingranaggio di Pascal, Leibniz e Babbage, ma con i progressi dell'elettronica dei primi anni Novanta questa barriera è stata superata. Tra gli esempi di quest'evoluzione ci sono la **macchina elettromeccanica** di George Stibitz, completata nel 1940 presso i Laboratori Bell, e Mark I, terminato nel 1944 all'Università di Harvard da Howard Aiken e da un gruppo di ingegneri dell'IBM (Figura

Augusta Ada Byron

Da quando il Dipartimento della Difesa degli Stati Uniti ha dato il suo nome a un linguaggio di programmazione, Augusta Ada Byron, contessa di Lovelace, è al centro di molte discussioni nella comunità informatica. Ada Byron ebbe una vita tragica, durata meno di 37 anni (1815-1852), che fu complicata dalla salute cagionevole e dal suo essere anticonformista in una società che limitava molto il ruolo professionale delle donne. Essa fu affascinata dalle creazioni di Charles Babbage nel 1833, quando partecipò a una dimostrazione di un prototipo della sua 'macchina alle differenze'. Il contributo di Ada Byron all'informatica comportò la traduzione dal francese all'inglese di un articolo che descriveva i progetti di Babbage riguardo alla macchina analitica. Per completare l'opera, Babbage la incoraggiò a scrivere

un'appendice in cui fossero descritte le applicazioni della macchina e che presentasse esempi su come programmarla per eseguire varie operazioni. L'entusiasmo di Babbage per il lavoro svolto da Ada (egli le diede numerosi suggerimenti e collaborò con lei all'elaborazione di alcuni esempi) era motivato dalla speranza che la sua pubblicazione portasse un sostegno finanziario alla creazione della macchina analitica. (Essendo la sorella di Lord Byron, Ada Byron aveva una certa fama, e quindi rapporti economici potenzialmente di prim'ordine). La sovvenzione non arrivò mai, ma l'appendice scritta da Ada Byron è giunta fino a noi ed è considerata una testimonianza d'interesse documentario circa i primi esempi di programmi per computer. Augusta Ada Byron è quindi riconosciuta oggi come il primo programmatore del mondo.

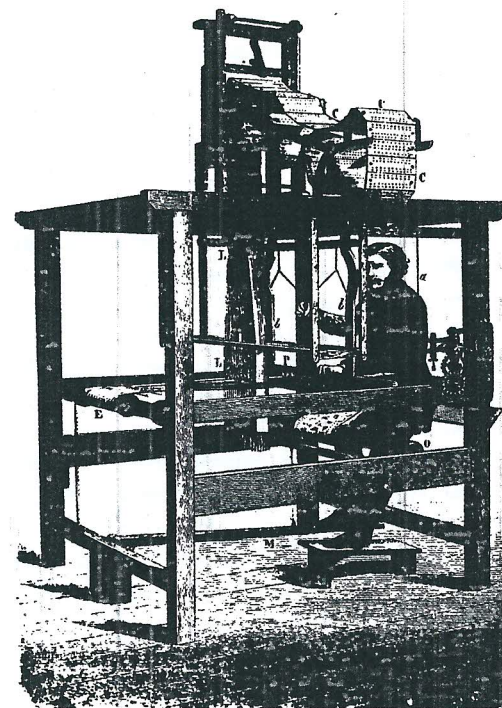


Figura 0.3 Telaio di Jacquard.

Courtesy of International Business Machines Corporation. Unauthorized use not permitted.

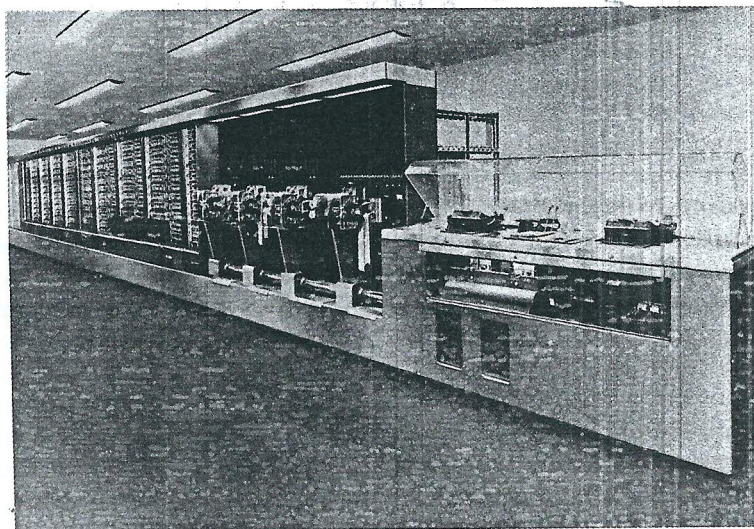
Macchina alle differenze di Babbage

I progetti della macchina di Charles Babbage anticiparono davvero la moderna progettazione dei computer. Se il livello tecnologico di allora avesse permesso di produrre le sue apparecchiature in modo economicamente conveniente, e se le esigenze di elaborazione dati delle imprese e della Pubblica Amministrazione fossero state della stessa portata di quelle di oggi, le idee di Babbage avrebbero potuto condurre alla rivoluzione informatica già nell'Ottocento. Invece, nel corso della sua vita, egli vide realizzato solo un modello dimostrativo della macchina alle differenze. Essa determinava valori numerici calcolando 'differenze successive'. Si può approfondire questa tecnica considerando il calcolo del quadrato dei numeri interi. Si inizia dalla nozione che il quadrato di 0 è 0, il quadrato di 1 è 1, quello di 2 è 4 e quello di 3 è 9. Da queste premesse si può calcolare il quadrato di 4 nel modo seguente. Prima di tutto si calcolano le differenze tra i quadrati che già si conoscono: $1^2 - 0^2 = 1$, $2^2 - 1^2 = 3$ e $3^2 - 2^2 = 5$; poi si calcolano le differenze tra questi risultati: $3 - 1 = 2$ e $5 - 3 = 2$. Si osservi che ambedue queste ultime differenze sono pari a 2. Supponendo che questa uniformità continui (si può dimostrare matematicamente che è così), si arriva alla conclusione che anche la differenza tra il valore $(4^2 - 3^2)$ e il valore

$(3^2 - 2^2)$ dev'essere 2. Pertanto $(4^2 - 3^2)$ dev'essere maggiore di $(3^2 - 2^2)$ di una quantità pari a 2. Ne consegue che $4^2 - 3^2 = 7$ e perciò $4^2 = 3^2 + 7 = 16$. Ora che si conosce il quadrato di 4, si può continuare la procedura per calcolare il quadrato di 5 in base ai valori di 1^2 , 2^2 , 3^2 e 4^2 . Anche se un esame più approfondito delle differenze successive va oltre lo scopo di questa descrizione, gli studenti di analisi matematica possono notare che l'esempio precedente si basa sul fatto che la derivata seconda di $y = x^2$ è una linea retta.

x	x ²	prima differenza	seconda differenza
0	0		
1	1	1	
2	4	3	2
3	9	5	2
4	16	7	2
5			2

Figura 0.4 Computer Mark I.



In tal senso esse erano già obsolete non appena costruite, perché altri ricercatori stavano nel frattempo usando la tecnologia dei tubi a vuoto allo scopo di costruire computer digitali totalmente elettronici. Il primo modello di computer di questo tipo fu la macchina di Atanasoff-Berry, costruita tra il 1937 e il 1941 presso l'Iowa State College (ora Iowa State University) da John Atanasoff e dal suo assistente, Clifford Berry. Un altro fu Colossus, creato in Inghilterra sotto la direzione di Tommy Flowers nell'ultimo scorcio della seconda guerra mondiale per decrittare i messaggi dei Tedeschi. (In realtà, di queste macchine ne vennero costruite di nascosto dieci, ma il segreto militare e i problemi di sicurezza nazionale imposero di tenerle fuori dall'albero genealogico dei computer). Presto seguirono altri sistemi più flessibili, come l'ENIAC (Electronic Numerical Integrator And Calculator, calcolatore e integratore numerico elettronico), sviluppato da John Mauchly e J. Presper Eckert alla Moore School of Electrical Engineering dell'Università della Pennsylvania.

Da questo momento in poi la storia delle macchine calcolatrici coincide in notevole misura con quella dello sviluppo tecnologico, le cui tappe sono state l'invenzione dei transistor e il successivo sviluppo di circuiti integrati, il lancio dei satelliti di comunicazione e i progressi nella tecnologia ottica. I computer da scrivania (desktop) attuali, e anche i loro cugini più piccoli, i portatili (laptop), hanno più potenza di calcolo delle macchine degli anni Quaranta, che avevano le dimensioni di una stanza, e possono scambiare informazioni rapidamente tramite sistemi di comunicazione globali.

Le origini di questi piccoli apparecchi si possono far risalire agli hobbisti che, dopo lo sviluppo di grandi macchine calcolatrici negli anni Quaranta, iniziarono a sperimentare con computer fatti in casa. Fu proprio nel tempo libero che Steve Jobs e Stephen Wozniak costruirono un computer domestico commerciabile, e nel 1976 fondarono la società Apple Computer per produrre e distribuire i loro prodotti. I prodotti Apple divennero famosi in breve tempo, ma non furono pienamente accettati dalla comunità delle aziende, che per la maggior parte delle sue esigenze informatiche continuava a rivolgersi all'ormai consolidata IBM.

Nel 1981, IBM presentò il suo primo computer da scrivania, chiamato personal computer, o, in forma abbreviata, PC. Il software del PC IBM fu sviluppato da una giovane azienda che stava cercando di affermarsi, la Microsoft. Il PC fu un successo immediato e legittimò il computer desktop come strumento di lavoro nelle imprese. Oggi il termine PC è ampiamente usato per tutti quei computer, di vari produttori, il cui progetto deriva dall'originale prodotto IBM. La maggior parte dei PC continua a essere commercializzata con il software Microsoft. A volte, tuttavia, la denominazione PC è utilizzata come sinonimo del termine generico desktop.

La disponibilità di computer da scrivania ha portato la tecnologia informatica in prima linea nel mondo contemporaneo. In realtà, la tecnologia informatica è così diffusa che saperla usare è essenziale per appartenere alla società moderna. È proprio tramite l'informatica che milioni di persone hanno accesso al sistema di connessione mondiale, l'Internet, un sistema che sta influenzando gli stili di vita e il commercio a livello mondiale.



0.3 Scienza degli algoritmi

Vincoli come una moderata capacità di memorizzazione dei dati e procedure di programmazione lente ed elaborate hanno limitato la complessità degli algoritmi che le prime apparecchiature erano in grado di eseguire. Non appena questi limiti furono superati, le macchine riuscirono però a svolgere operazioni sempre più complicate. Quando i tentativi di esprimere in forma algoritmica i compiti da affidare alle macchine incominciarono a mettere alla prova le capacità della mente umana, si investirono sempre più risorse nella ricerca indirizzata allo studio degli algoritmi e al processo di elaborazione.

Fu in questo contesto che il lavoro teorico dei matematici iniziò a dare i suoi frutti. Conseguentemente al teorema di incompletezza di Gödel, i matematici avevano da tempo preso in considerazione le questioni relative ai processi algoritmici che la tecnologia informatica iniziava a porre. C'erano quindi tutti i presupposti perché emergesse una nuova disciplina: l'*informatica*.

Oggi questa disciplina si è universalmente affermata come la scienza degli algoritmi. Come si è visto, il suo ambito è molto esteso, in quanto attinge da discipline diverse: la matematica, l'ingegneria, la psicologia, la biologia, l'amministrazione aziendale e la linguistica. Molti temi oggetto di studio da parte dell'informatica saranno trattati nel capitolo successivo; in ogni caso, l'obiettivo di questo libro è presentare i concetti principali, i temi di ricerca e alcune tecniche utilizzate per aumentare le conoscenze in questo ambito.

È importante distinguere tra informatica e applicazioni informatiche, analogamente a quanto si fa tra fisica e ingegneria meccanica. La fisica è la scienza che cerca di spiegare il rapporto tra forza, massa e accelerazione; l'ingegneria meccanica è l'applicazione di questa scienza. Un ingegnere meccanico deve capire la fisica, ma a seconda della propria specialità deve anche conoscere i vari gradi di purezza dell'acciaio, la composizione del calcestruzzo oppure la disponibilità sul mercato di cuscinetti a sfera. Al contrario, un fisico è interessato alle discrepanze delle teorie di Newton e a come sono state risolte dalle intuizioni di Einstein. Non ci si dovrebbe aspettare, quindi, che chi studia fisica si occupi delle varie pulegge disponibili sul mercato o delle regolamentazioni ministeriali che riguardano le cinture di sicurezza sulle automobili.

In modo simile, questo libro non conterrà le istruzioni per usare un particolare programma applicativo oppure per installare un browser. Questo non significa che tali argomenti siano irrilevanti, solo non fanno parte dell'informatica. Analogamente, l'esame generale della programmazione non si concentrerà qui sullo sviluppo delle competenze tecniche in un particolare linguaggio. Verterà, invece, sui principi che stanno alla base degli strumenti di programmazione attuali, sulla loro evoluzione e su problemi che la ricerca sta cercando di superare.

Mentre si procede in questo approfondimento è facile perdere di vista la visione generale. È meglio, quindi, fare una pausa per raccogliere le idee e mettere a fuoco alcune domande che consentono di definire la scienza informatica individuandone gli argomenti centrali.

- ◆ Quali problemi si possono risolvere con gli algoritmi?
- ◆ Come si può semplificare la messa a punto di nuovi algoritmi?

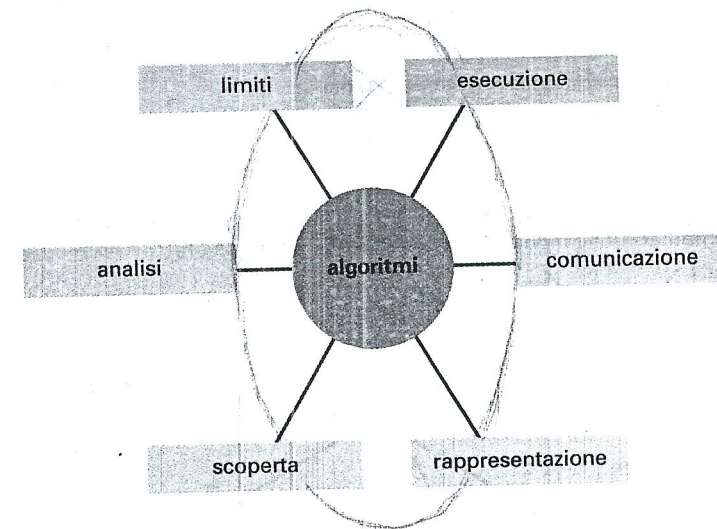


Figura 0.5 Ruolo centrale degli algoritmi in informatica.

- ◆ Come si possono migliorare le tecniche di rappresentazione e di comunicazione degli algoritmi?
- ◆ Come si possono applicare le nostre conoscenze degli algoritmi e della tecnologia per ottenere macchine migliori?
- ◆ Come si possono analizzare e confrontare le caratteristiche di algoritmi diversi?

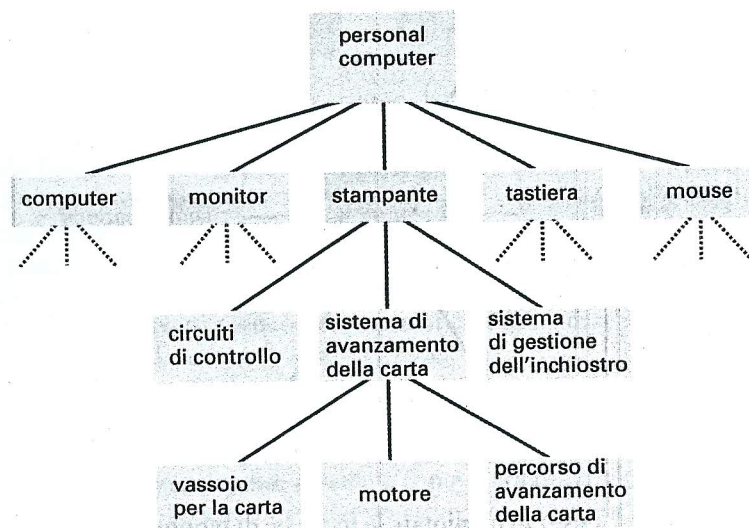
Si noti che il denominatore comune di tutti questi quesiti è lo studio degli algoritmi (Figura 0.5).

0.4 Ruolo dell'astrazione

Il concetto di astrazione è così pervasivo nello studio dell'informatica e nel progetto di sistemi di calcolo che va affrontato già in questo capitolo preliminare. Secondo l'accezione usata in questo libro, **il termine astrazione si riferisce alla distinzione tra le proprietà esterne di un'entità e i dettagli della sua struttura interna**. È l'astrazione che permette di ignorare i particolari interni di un congegno complesso come un computer, un'automobile o un forno a microonde, e di usarlo come una singola unità. Inoltre, è sempre tramite l'astrazione che queste macchine complesse vengono progettate. Un'automobile, per esempio, è progettata in modo gerarchico. Al livello più alto è vista come un insieme di grandi componenti – motore, sistema di sospensioni e trasmissione –, senza considerare i dettagli interni di ciascun sottosistema. Ogni componente è a sua volta costruito partendo da vari componenti, e così via.

L'hardware di un tipico personal computer ha una struttura gerarchica analoga (Figura 0.6). Al livello più alto può essere considerato come un insieme di macrocomponenti: il computer stesso, una tastiera, un monitor, un mouse e una stampante. Se ci si focalizza sulla stampante, si scopre che è formata da componenti più piccoli, come il meccanismo di avanzamento della carta, il sistema logico di controllo e quello di disper-

Figura 0.6 Gerarchia di astrazione nell'hardware di un tipico personal computer.



sione dell'inchiostro. Il meccanismo di alimentazione della carta consiste di unità ancora più piccole: il vassoio per la carta, il motore elettrico e il percorso di avanzamento carta.

L'astrazione consente di costruire, analizzare e gestire grandi sistemi complessi, che non sarebbero comprensibili nella loro interezza se considerassimo i minimi particolari. Di fatto, applicando l'astrazione ci si può avvicinare a questi sistemi a diversi livelli di dettaglio. A ogni livello si descrive il sistema in termini di componenti, definiti **strumenti astratti**, di cui si ignora la composizione interna. Ciò permette di concentrare l'attenzione sull'interazione di ogni componente con gli altri dello stesso livello e su come l'intero insieme di componenti, considerato un **tutt'uno**, costituisce un componente di livello superiore. In questo modo è possibile cogliere la parte del sistema che ci interessa per comprendere una certa operazione senza perdersi in un mare di minuzie. Questa è la bellezza dell'astrazione.

Va detto che l'astrazione non si limita alla scienza e alla tecnologia. È un'importante tecnica di semplificazione grazie alla quale la società ha creato uno stile di vita che altrimenti sarebbe impossibile. Poche persone sono in grado di capire come sono realizzate le varie comodità della vita moderna. Si mangiano cibi e si indossano vestiti che non si sarebbe capaci di produrre da soli. Si utilizzano dispositivi elettrici senza capirne la tecnologia sottostante. Si fruisce di servizi forniti da altri senza conoscere i dettagli delle loro professioni. A ogni nuovo progresso della scienza, una piccola parte della società sceglie di investirlo nella realizzazione di risultati concreti, mentre il resto impara a utilizzare questi risultati come strumenti astratti. In questo modo, il patrimonio di strumenti astratti dell'umanità cresce e le capacità di evoluzione della società aumentano ulteriormente.

Anche questo libro applica l'astrazione per formare capitoli (e anche paragrafi all'interno dei capitoli) sorprendentemente indipendenti. Gli argomenti trattati all'inizio del

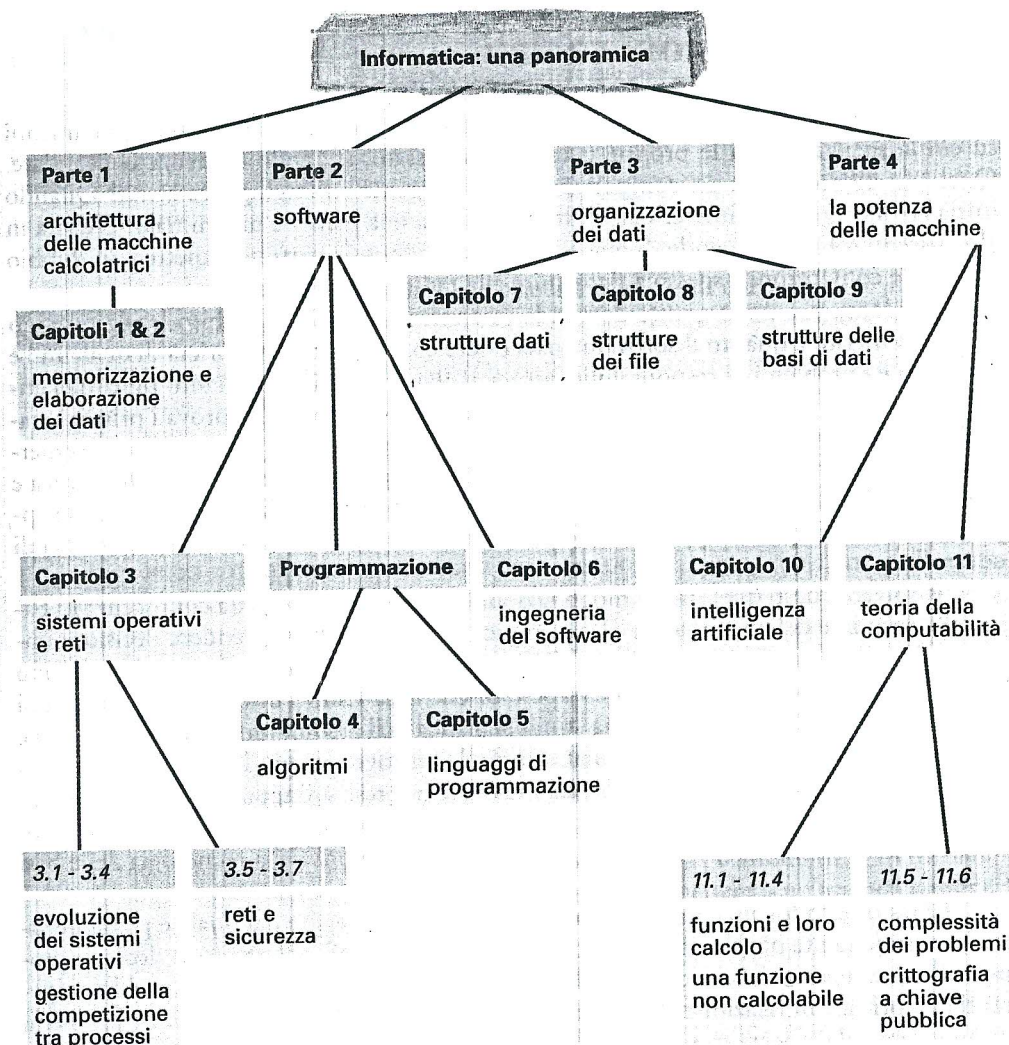


Figura 0.7 Visualizzazione di questo testo come gerarchia di strumenti astratti.

testo fungono da strumenti astratti per quanto segue: pertanto non è indispensabile una perfetta comprensione dei primi capitoli per capire gli ultimi. È possibile, per esempio, iniziare a studiare dal Capitolo 10 (Intelligenza artificiale) e capire ugualmente buona parte degli argomenti trattati. (Se si è dimenticato il significato di un termine tecnico si può sempre utilizzare l'indice). Nello stesso spirito della Figura 0.6, la Figura 0.7 illustra la composizione gerarchica di questo testo, che è articolato in quattro parti affrontabili, in grande misura, autonomamente. All'interno di ogni parte ci sono poi gruppi di paragrafi e capitoli che possono essere studiati come unità di astrazione ancora più piccole. Questa divisione non è soltanto un prodotto del testo, rispecchia la struttura della scienza. In effetti, l'informatica è costituita da molti argomenti distinti, ma correlati tra loro.

0.5 Ripercussioni sociali

I progressi della scienza e della tecnologia hanno messo in discussione molte distinzioni su cui la società basava le proprie decisioni in passato, sfidando credenze consolidate. Qual è la differenza tra un comportamento intelligente e l'intelligenza stessa? Quando inizia la vita? Quando finisce? Che differenza c'è tra una pianta e un animale? C'è vita in altri sistemi solari? Domande di questo tipo costringono le persone a mettere in dubbio le proprie convinzioni e spesso a ricostruire alla radice le loro certezze.

L'informatica pone queste domande in contesti diversi. In ambito legale sorgono quesiti in merito a quanto il software possa essere considerato di proprietà di qualcuno e sui diritti e i problemi di responsabilità associati alla proprietà. Per quanto riguarda l'etica, gli utenti si trovano di fronte a numerose scelte che mettono alla prova i principi tradizionali su cui si basa il loro comportamento. In campo politico le questioni che emergono riguardano i limiti di diffusione che andrebbero posti alla tecnologia informatica e alle sue applicazioni. Un dubbio sempre più insistente pervade la società: se le recenti applicazioni dei computer rappresentino una nuova forma di libertà o una nuova forma di controllo.

Risolvere questi dilemmi in modo razionale richiede un'adeguata conoscenza di base della scienza e della tecnologia. Per esempio, se la società deve prendere decisioni importanti circa l'immagazzinamento e l'eliminazione di scorie nucleari, i cittadini devono conoscere gli effetti della radiazione, sapere di che cosa hanno bisogno per proteggersi dai danni che ne derivano e fare una stima realistica dell'arco di tempo durante il quale persisterà il rischio di essere contaminati. In modo analogo, per giudicare se dovrebbe essere permesso alla pubblica amministrazione o alle imprese sviluppare grandi basi di dati integrate che contengono informazioni sui cittadini o sui clienti bisogna avere una conoscenza basilare delle capacità e dei limiti delle implicazioni della tecnologia delle basi di dati.

Questo testo fornisce un background importante, a partire dal quale è possibile affrontare seriamente problemi di questo tipo. Alcuni paragrafi, infatti, sono dedicati proprio alle questioni sociali, etiche e legali. Vengono esaminati per esempio i problemi inerenti alla privacy in relazione alla tecnologia di Internet e delle basi di dati, e i problemi di proprietà del software in relazione allo sviluppo del software. Anche se non sono esattamente parte dell'informatica, questi argomenti risultano centrali per chi progetta la propria carriera in campi limitrofi a quello informatico.

La sola conoscenza tecnica non fornisce necessariamente la soluzione a molti problemi creati dagli attuali progressi dell'informatica. Spesso non esiste un'unica risposta corretta, e molte soluzioni valide sono il risultato del compromesso tra punti di vista opposti. Trovare soluzioni richiede quindi la capacità di ascoltare, riconoscere altri punti di vista, sostenere una discussione razionale e cambiare le proprie opinioni quando si acquisiscono nuove conoscenze. Tenendo presente tutto ciò, ogni capitolo termina con una serie di domande intitolate 'Questioni sociali'. Non si tratta di quesiti cui dare per forza una risposta, ma di problematiche da prendere in considerazione. In molti casi una risposta che a prima vista può sembrare logica diventa insoddisfacente non appena si esaminano le alternative. Chiudo questa introduzione con una serie di domande che riguardano l'informatica in generale.

Questioni sociali

Le domande che seguono sono poste per aiutarvi a capire alcuni problemi etici, sociali e legali associati al campo dell'informatica. L'obiettivo non è soltanto che rispondiate a questi quesiti, ma anche che consideriate perché avete dato una certa risposta e valutate se le vostre giustificazioni sono coerenti passando da una domanda alla successiva.

1. La premessa che la società è *diversa* da come sarebbe stata senza la rivoluzione informatica è generalmente accettata. La società è *migliore* di come sarebbe stata senza questa rivoluzione? È *peggiore*? La risposta dipende dalla vostra posizione sociale?
2. È giusto far parte della società contemporanea senza fare un minimo sforzo per capire gli elementi base della tecnologia? Per esempio i cittadini di un Paese democratico, il cui voto spesso decide l'impiego della tecnologia, hanno l'obbligo di cercare di comprenderla? La risposta dipende dal tipo di tecnologia considerata? La risposta è la stessa se si tratta, per esempio, di tecnologia nucleare o di tecnologia informatica?
3. Se si usa denaro contante nelle transazioni finanziarie, di solito si possono gestire affari senza sovrapprezzi di servizio. Tuttavia, poiché la maggior parte dell'economia si sta automatizzando, le società finanziarie addebitano spese di servizio ai clienti che accedono ai sistemi automatici di commercio e finanza. Si può sostenere che questi addebiti limitano ingiustamente l'accesso degli individui all'economia? Si supponga, per esempio, che un datore di lavoro paghi i dipendenti solo con assegni, e tutte le banche applichino addebiti sul prelievo e sul deposito tramite assegni: i lavoratori sono trattati in modo corretto? Che cosa accade se un datore di lavoro insiste per pagare soltanto tramite versamento diretto?
4. Nel contesto della televisione interattiva, fino a che punto un'azienda dovrebbe poter recuperare informazioni sulle famiglie interpellando i bambini (magari mediante un gioco interattivo)? Si dovrebbe, per esempio, permettere a un'azienda commerciale di ottenere un resoconto sui modelli di acquisto dei propri genitori da parte di un bambino? Che cosa dire delle informazioni sul bambino stesso?
5. Fino a che punto un governo dovrebbe regolare per legge la tecnologia informatica e le sue applicazioni? Si considerino, per esempio, i problemi menzionati nelle Domande 3 e 4. Che cosa prevede la legislazione in materia?
6. Fino a che punto le attuali decisioni sulla tecnologia in generale, e sulla tecnologia informatica in particolare, influenzeranno le generazioni future?
7. Man mano che la tecnologia progredisce, il sistema scolastico è costantemente impegnato a riconsiderare il livello di astrazione con cui vengono presentati gli argomenti. Si discute molto sul fatto se sia ancora necessaria la competenza tecnica, oppure se si debba consentire agli studenti di basarsi solo su nozioni astratte. Agli studenti di trigonometria non si insegna più come calcolare i valori delle funzioni trigonometriche con le tabelle; per ottenere tali valori, essi utilizzano le calcolatrici come strumenti astratti. Alcuni sostengono che anche la divisione dovrebbe essere abbandonata. Quali altri argomenti si prestano a simili controversie? L'impiego della tecnologia video eliminerà un giorno il bisogno di leggere? I controllori ortografici automatici rendono inutile conoscere le regole dell'ortografia?

8. Si supponga che il concetto di biblioteca pubblica si fondi sulla premessa che in una democrazia tutti i cittadini devono avere accesso alle informazioni. Considerato che le informazioni sono sempre più memorizzate e diffuse per mezzo della tecnologia informatica, l'accesso a quest'ultima diventa un diritto di ogni cittadino? Se sì, le biblioteche pubbliche dovrebbero essere il canale attraverso il quale favorire tale accesso?
9. Quali problemi etici si pongono in una società che si basa sull'uso di strumenti astratti? Esistono casi in cui non è etico usare un prodotto o un servizio senza sapere in che modo è realizzato? Oppure senza capire le conseguenze del suo utilizzo?
10. Dal momento che l'economia è sempre più automatizzata, lo stato controlla più facilmente le operazioni finanziarie dei suoi cittadini. È un fatto positivo o negativo?
11. Quali tecnologie immaginate da George Orwell (Eric Blair) nel suo romanzo 1984 sono diventate effettivamente realtà? Sono usate come Orwell aveva previsto?
12. Nel tentativo di analizzare le scelte etiche, i ricercatori nel campo della filosofia morale hanno sviluppato varie teorie. Una teoria molto diffusa afferma che le decisioni dovrebbero basarsi sulle conseguenze. All'interno di questa sfera c'è, per esempio, l'utilitarismo, secondo cui la decisione 'corretta' è quella che genera il maggior vantaggio per il numero più elevato di persone. Un'altra teoria si basa sui doveri anziché sulle conseguenze, e sostiene che la decisione 'corretta' è quella che rispetta i diritti dei singoli. Le risposte date alle domande precedenti indicano che tendete a seguire l'etica basata sui doveri o quella basata sulle conseguenze?

Lecture consigliate

- Dejoie D., G. Fowler, D. Paradice, *Ethical Issues in Information Systems*, Boston, Boyd and Fraser, 1991.
- Edgar S. L., *Morality and Machines*, Sudbury, MA, Jones and Bartlett, 1997.
- Forester T., P. Forrison, *Computer Ethics: Cautionary Tales and Ethical Dilemmas*, Cambridge, MA, MIT Press, 1990.
- Goldstine J. J., *The Computer from Pascal to von Neumann*, Princeton, Princeton University Press, 1972.
- Johnson D. G., *Computer Ethics*, seconda ed., Englewood Cliffs, NJ, Prentice-Hall, 1994.
- Johnson D. G., *Ethical Issues in Engineering*, Englewood Cliffs, NJ, Prentice-Hall, 1991.
- Kizza J. M., *Ethical and Social Issues in the Information Age*, New York, Springer-Verlag, 1998.
- Mollenhoff C. R., *Atanasoff: Forgotten Father of the Computer*, Ames, Iowa State University Press, 1988.
- Neumann P. G., *Computer Related Risks*, Reading, MA, Addison-Wesley, 1995.
- Randell B., *The Origins of Digital Computers*, New York, Springer-Verlag, 1973.
- Rosenoer J., *CyberLaw: The Law of the Internet*, New York, Springer-Verlag, 1997.
- Shurkin J., *Engines of the Mind*, New York, Norton, 1984.
- Spinello R. A., H. T. Tavani, *Readings in CyberEthics*, Sudbury, MA, Jones and Bartlett, 2001.
- Woolley B., *The Bride of Science, Romance, Reason and Byron's Daughter*, New York, McGraw-Hill, 1999.